Scan Detection on Very Large Networks Using Logistic Regression Modeling

Carrie Gates, Joshua J. McNutt, Joseph B. Kadane, and Marc I. Kellner
Carnegie Mellon University,
Pittsburgh, PA 15213, USA
{cgates,jmcnutt,kadane,mik}@cmu.edu

Abstract

*Scanning activity is a common activity on the Internet today, representing malicious activity such as information gathering by a motivated adversary or automated tools searching for vulnerable hosts (*e.g.*, worms). Many scan detection techniques have been developed; however, their focus has been on smaller networks where packet-level information is available, or where internal characteristics of the network are known. For large networks, such as those of ISPs, large corporations or government organizations, this information might not be available. This paper presents a model of scans that can be used given only unidirectional flow data. The model uses a Bayesian logistic regression, which was developed using a combination of expert opinion and manually-classified training data. It is shown to have a detection rate of 95.5% with a false positive rate of 0.4% overall when tested against a set of 300 TCP events.*

1 Introduction

A scan is a reconnaissance technique aimed at multiple targets. The goal of a scan is to determine the presence of particular hosts, or particular services on particular hosts. It consists of sending a probe packet to the particular target, where the response from the target indicates if the host and/or service is present.

Scans are often indicators of malicious activity. They can indicate the presence of particular worms, as the majority of worms observed to date have been scanning worms [15]. Scans, particularly those generated by vulnerability scanners, can also indicate that an adversary is about to perform an attack, as was investigated by Panjwani *et al.* [9]. Finally, an attack itself can appear as a scan to a defender, where the adversary might be attempting a particular exploit against all of the addresses on a particular subnet.

As scans can indicate malicious activity, it is important to detect their presence. However, most scan detection approaches have focused on detecting scans on smaller networks, where the characteristics of the network are entirely known or where packet-level traffic information is available. This paper presents a novel approach to scan detection that addresses the issue of detecting scans on a large network, such as an ISP, where only unidirectional flow-level information is available.

This paper is organized as follows: Section 2 provides some background on scanning activity and other approaches to detecting scans. Section 3 introduces the system requirements and constraints that arise from dealing with large networks. Our model of scans is presented in Section 4. In Section 5 we compare our results to that of the Threshold Random Walk (TRW) algorithm [6]. Section 6 provides some concluding remarks.

2 Background

Several researchers have examined the detection of port scans, resulting in a variety of definitions. In general, a scan is a reconnaissance technique used to determine the existence of multiple targets, where the targets could be hosts or particular services on particular hosts. Scans can be used both by adversaries in order to determine what to attack and by system administrators to audit their network. Scans also occur from system misconfigurations, as well as from peer-to-peer applications searching for previously-contacted peers. Scans can be a side effect of vulnerability searches, such as by vulnerability scanners or worms.

Two popular intrusion detection systems — Snort [12] and Bro [10] — use thresholding to detect scanning activity. However, both approaches have been found by Jung *et al.* [6] to have low values for efficiency and/or effectivness. Bro's scan detection has since been modified to incorporate a threshold random walk approach [6].

Various algorithmic approaches to detecting scans have also been developed. For example, Graph-based Intrusion Detection System, GrIDS, recognizes scans based on structures the communications form when inserted into a graph [13]. Leckie and Kotagiri [7] use probabilistic modeling to determine how likely it is that a source will contact a particular destination IP address or port, using the conditional probabilities to determine if a source is scanning. Robertson *et al.* [11] developed a method based on the traffic returned to a source, where no response or a RST-ACK was indicative of scanning. Approaches based on visual representation of connection data have also been developed for detecting port scans (*e.g.*, Muelder *et al.* [8]).

However, the threshold random walk (TRW) approach to detecting port scans developed by Jung *et al.* [6] has become the gold standard for scan detection, and has been used for activities such as worm detection and quarantine [14]. Their approach uses sequential hypothesis testing, where each new connection request from an external source is evaluated. If the destination exists, then there is more support for the source being benign (or, rather, not scanning). However, if the destination does not exist, then there is more support for the source to be scanning. Once the hypothesis that the source is scanning has been accepted or rejected, it is labeled with the result. The use of sequential hypothesis testing allows the user to customize variables based on the density of hosts on their network and on the desired detection and false positive rates.

3 System Requirements and Constraints

Focusing on very large networks introduces a number of system requirements and constraints that are unique to such networks. In particular, we operate under the following assumptions of operating conditions:

- Only flow level traffic collection, such as Cisco NetFlow, will be performed. This assumption stems from the large volume of traffic such networks typically observe. As a result, the cost of collecting packet headers is typically prohibitive, both in terms of the cost to the router performing the collection and in terms of the storage requirements. Therefore sites often use flow data as an alternative.

- There will be multiple border routers. Thus traffic will be collected at multiple locations and need to be aggregated at a single collection site.

- There will be multiple geographic and administrative domains. Given that we are working at the ISP level, it is likely that the network will span multiple countries. Additionally, the subnets within the ISP will represent multiple administrative domains, where the ISP might not have visibility into how the subnets have been configured or what policies might be in place.

- Any flow data collected will be unidirectional. This is the case with Cisco NetFlow, which is a popular flow collection format. Additionally, a large organization might have asymmetric routing policies so that bidirectional flow information can not be collected.

Current approaches to scan detection do not work given all of the above conditions.

4 Scan Model

Given that a scan consists of a collection of activity originating from a single source, regardless of the type of scan being performed, we choose to analyse all of the flows collected from a single source IP address as an event. We define an event as a collection of flows originating from the same source surrounded by periods of inactivity. For the purposes of our analysis, we define an event as consisting of a minimum of 32 flows, all with the same protocol, surrounded by at least 5 minutes where no flows are observed, where the values of 32 and 5 have been chosen arbitrarily.

Each event can now be analysed to determine if it contains a scan. Given the boolean nature of the decision — yes, the event contains a scan, or no, it does not — a Bayesian logistic regression approach is used to model the information that informs a user if a scan is present. There are three advantages to using a Bayesian logistic regression in these circumstances: (1) the model will return a probability that an event contains a scan, allowing the user to chose cut-offs that balance their tolerance for false positives versus false negatives, (2) the logistic regression learns a model based on labeled data rather than using general heuristics, and (3) the Bayesian approach uses both expert opinion and observed data.

For the TCP protocol, we identified 21 different variables as being possible indicators of whether the event contained scanning activity. These variables are:

1. maximum /24 subnet run length,

2. ratio of flows that do not have the ACK bit set to all flows,

3. ratio of flows to known malware ports to all flows,

4. ratio of flows with fewer than 3 packets to all flows,

5. maximum run length of IP addresses in any one /24 subnet,

6. maximum number of IP addresses contacted in any one /24 subnet,

7. maximum number of high destination ports contacted on any one host,

8. maximum number of low destination ports contacted on any one host,

9. maximum number of consecutive high destination ports contacted on any one host,
10. maximum number of consecutive low destination ports contacted on any one host,
11. number of unique destination IP addresses,
12. number of unique source ports,
13. average number of source ports per destination IP address,
14. ratio of flows with “standard” flag combinations (SYN and ACK set, along with either the FIN or RST bit set) to all flows,
15. ratio of the number of flows with the average bytes/packet > 60 to all flows,
16. median value of packets per destination IP address,
17. ratio of flows with “standard” combination (standard flag combination and at least three packets and at least 60 bytes/packet on average) to all flows ,
18. ratio of flows with backscatter combination (RST, RST-ACK, or SYN-ACK for the flag combination and the average number of bytes/packet is ≤ 60 and the number of packets per flow is ≤ 2) to all flows,
19. ratio of unique destination IP addresses to number of flows,
20. ratio of unique source ports to number of flows, and
21. ratio of flows with backscatter flag combinations (SA—RA—R) to all flows.

In some cases the variables were chosen because we suspected that a high value would be a good indication of scanning activity (*e.g.*, the ratio of flows that do not have the ACK bit set to all flows), while in other cases the variables were chosen because a high value indicated that the event probably did *not* contain scanning activity (*e.g.*, ratio of flows with backscatter flag combinations to all flows).

4.1 Model Development

The classical approach to developing a logistic regression model consists of choosing an appropriate training set and then generating the model based on the data in the training set. However, we use instead a Bayesian approach to logistic regression modeling. The Bayesian approach seeks to assign priors to each of the co-efficients based on expert opinion of the contribution each variable makes. A prior can be loosely thought of as a weighting on the co-efficients. This process is followed in order to develop a model that is based on a combination of expert opinion and the models generated by the data itself. It has the advantages of requiring less training data and can reduce errors caused by choosing a training set that is not representative of the entire data space.

Two data sets were gathered — one for the elicitation process and one for the training process. Events were generated based on flow data collected during a one hour period on a large network. The elicitation period was 17:00-18:00 GMT on May 4, 2005, where 129,191 events were collected. The training set consisted of 130,062 events gathered during the subsequent hour. The values for each of the 21 variables were calculated for each event in both sets. These values were then used to choose appropriate subsets from each set for manual inspection.

We made the assumption that each variable would have a linear relationship with the dependent variable, where the dependent variable is the probability that an event represents scanning activity. Based on this assumption, the values that provide the most information in determining appropriate co-efficients for a general linear model are the values located at the extremes [3]. This is because the variance in the estimated dependent variable $\hat{y}$ is proportional to the variance of each of the co-efficients, and so reducing the variance of the co-efficients also reduces the variance in $\hat{y}$. This is achieved by maximizing the value for the sum of squares for each variable, since there is an inverse relationship between the sum of squares and the variance. The sum of squares is maximized by choosing values that are furthest from the mean [16, p. 13-15], and so the extreme values are chosen for generating the model.

Our aim was to identify the 100 most extreme observations in X-space. With 21 variables, we needed to seed our X matrix with at least 21 observations so that it is invertible. Thus each row in X represents a single observation, while each column represents a single variable. We use the fact that the variance of any additional observation v is proportional to $v^T(X^TX)^{-1}v$, where v^T is the transpose of the vector v, X^T is the transpose of matrix X and $(X^TX)^{-1}$ is the inverse of the matrix X^TX. Therefore, we randomly select observations to include in our matrix X until we have enough cases to invert the matrix. For our experiments, we arbitrarily chose 30 samples to initially seed our matrix X. Then, for each remaining observation not yet selected, we calculate its variance. The next observation added to X will be the observation with highest variance. This procedure is repeated until X contains 100 rows, and each new set of calculations are performed using the new, larger X matrix.

For the elicitation set, the first 100 cases in the X matrix (so 30 randomly chosen observations plus the 70 observations with the largest variance) were selected for manual analysis. An analyst was provided with the values for each of the 21 variables, but was not allowed to use any additional information. Based on these values, she provided her estimate of the probability that the event contained a scan. These probabilities provided the values for y used to then determine the priors for the logistic regression. The priors are calculated by first converting each probability y_i onto the logit scale $log(\frac{y_i}{1-y_i})$ and then using a linear regression to determine the prior values for each co-efficient.

The training set consisted of 200 observations chosen using the same procedure that was used to select the elicitation set. In this instance, however, a manual analysis was performed using the event data itself. All of the flows were provided to the expert, who flagged each event with a boolean value indicating if the event contained a scan. Of the 200 observations, 53 were flagged as containing scanning activity. A Bayesian logistic regression model was generated that incorporated the priors obtained during the elicitation process. The posterior distribution for each of the co-efficients is proportional to the product of the sampling distribution, or likelihood, and the prior distribution for each observation [4, p. 65-88]. This process assumes that the priors for the co-efficients follow a normal distribution, which is generally not the case. Therefore the values required were calculated using a Markov Chain Monte Carlo simulation [5]. The result is a logistic regression model where the posterior co-efficients have been informed by both expert option (via the priors) and the sample training data.

Given that the resulting logistic regression model contains 21 variables we wanted to reduce the number of variables in order to reduce the overhead associated with calculating the values for each of the different variables and to therefore reduce the processing time for each event. We used the Akaike Information Criterion (AIC) [1] to determine what variables could be removed without significantly affecting the model's fit to the data. The final result was the following model, consisting of six variables:

$$\hat{P}(\text{event contains a scan}) = \frac{e^{\hat{y}}}{1+e^{\hat{y}}}$$

where

$$\hat{y} = -2.83835 + 3.30902x_2 - 0.15705x_4 - 0.00232x_{13} - 1.04741x_{15} + 3.16302x_{19} - 3.26027x_{21}$$

where x_2 is the ratio of flows with no ACK bit set to all flows, x_4 is the ratio of flows with fewer than three packets to all flows, x_{13} is the average number of source ports per destination IP address, x_{15} is the ratio of the number of flows that have an average of 60 bytes/packet or greater to all flows, x_{19} is the ratio of the number of unique destination IP addresses to the total number of flows, and x_{21} is the ratio of the number of flows where the flag combination indicates backscatter to all flows.

4.2 Model Validation

This model was validated using a third set of data. This data was collected from the same network on May 4, 2005, from 19:00-20:00 GMT. This hour contained 127,873 events, from which 300 events were drawn randomly. These 300 events were analysed by the same expert who analysed the elicitation and training set. The expert was provided with the flow data for each of the events, and asked to label each event as either containing a scan or not.

Each of the 300 events were evaluated by the logistic regression model. Given that the model returns a probability that the event contained a scan, we set the threshold so that anything greater than or equal to 0.5 was considered a scan, while anything less than that was not. These values were compared to the expert assessment. Of the 300 events, there were 22 scans and 278 non-scans. The model correctly recognized 21 scans and 277 non-scans. The detection rate and false positive rate, using the conditional probability definitions provided by Axelsson [2], are therefore 95.5% and 0.4% respectively. The one false positive had a probability of being a scan of 60.74% — the lowest probability of any of the scans. This non-scan consisted of 62 flows to three destinations. Two flows consisted of completed connections to two of the addresses. The remaining 60 flows were to a single IP address and consisted of four packets with a SYN-RST flag combination. The one false negative had a probability of being a scan of 41.04% — the highest probability of any of the non-scans. This scan consisted of 70 flows to seven unique destination IP addresses, all of which were one to three packet SYN-only flows.

4.3 Analysis of Variables

The sign of the co-efficients indicates if the probability of a scan increases or decreases as the value of the variable increases. Thus as variables x_2 (the ratio of flows with no ACK bit set to all flows) and x_{19} (the ratio of the number of unique destination IP addresses to the total number of flows) increases, so too does the probability that the event being examined contains a scan. This result is intuitive in both cases. In the first case, the ACK flag generally indicates that information has been exchanged during a communication, whereas a high proportion of flows where the ACK flag is not present indicates communication attempts that lack any connection and exchange of information. In the second case, a high ratio of destination IP addresses to the number of flows indicates that the source IP address is communicating with a large number of different destination IP addresses, but with few flows to each destination. In isolation, this does not necessarily indicate scanning behaviour (*e.g.*, external web servers might display similar connection patterns), however it can be a good indicator when combined with the other indicators, such as the proportion of flows with the ACK bit set (*e.g.*, web server traffic would largely have the ACK bits set indicating completed connections, whereas scan traffic likely will not).

Four of the variables have negative co-efficients, indicating that as their value increases, the probability that the event contains a scan decreases. These variables are: the ratio of flows with fewer than three packets to all flows (x_4), the average number of source ports per destination IP address (x_{13}), the ratio of flows with an average of 60 bytes/packet or greater to all flows (x_{15}), and the ratio of flows with a backscatter flag combination to all flows (x_{21}). It is not surprising that a high value for x_{13} indicates that a scan is less likely, since a high number of source ports per destination IP address indicates multiple connections to the same machine, which is more likely to occur given normal communications then it is during horizontal or strobe scans. A high value for x_{15} indicating normal activity is also not surprising, given that finding 60 bytes/packet or more indicates that data was likely exchanged. Given that a blind scan has a low likelihood of finding machines running the target service, the majority of flows will not show that data was exchanged. A high value for x_{21} indicates that the majority of flags have backscatter combinations (SYN-ACK, RST-ACK or RST). Finally, variable x_4 is surprising. This variable indicates that if a large number of flows have fewer than three packets then it is less likely to be a scan. We believe that this is the case for two reasons: (1) that a high value here was often observed to be associated with backscatter traffic, which is a common occurrence on our network, and (2) that a large number of scans use the TCP stack implementation where a lack of response to a SYN results in two more attempts before considering the connection to have failed.

5 Comparison to TRW

Given that the Threshold Random Walk (TRW) [6] is the current gold standard for detecting port scans, we wanted to use it for a basis of comparison. However, TRW requires either bi-directional flow information (so that the response to a connection request is known) or an oracle that knows if an internal host exists. Under our operating conditions, neither is available.

We were able, however, to modify the TRW algorithm to perform a two-step process that emulates having an oracle available. For a given hour all of the outgoing data was used to create a set containing every source IP observed. The assumption is that if a source IP address was observed in the outgoing traffic, then that host exists, otherwise it does not. The TRW algorithm was then used on the incoming traffic, with this set of IPs used for the oracle. We used the same values from the original paper — $\theta_0 = 0.8$, $\theta_1 = 0.2$, $\alpha = 0.01$, and $\beta = 0.99$. This assumes that the network density is 20%, and that the false positive rate should be less than 1%. Given that we use unidirectional flow data, we do not know if the external source originated the conversation. Additionally, we know that there are backdoors on our network that have not been instrumented, and so our list of known existing hosts is not complete. In order to avoid any false positives that these constraints might generate, the TRW algorithm was further modified so that a flow to a non-existing destination was considered a miss only if both the destination IP address was not in the set of hosts known to exist **and** the flag combination for the flow was a SYN only.

We compare the results for TRW and our model on both the training set and testing set. This was done because the data for the testing set was randomly chosen, whereas the data for the training set was specifically chosen to represent the more extreme cases. Of the 200 observations in the training set, 53 were identified as scans via expert analysis. Our model correctly recognized all 53, however it also had three false positives. The probabilities of the event containing a scan for each of the three false positives were all between 50% and 60%. Given the same set, TRW correctly identified 50 of the 53 scans, with only one false positive. The three scans that were not identified by TRW were: (1) a vertical scan of a single IP address, (2) a horizontal scan of port 53 on 14 IP addresses, all of which were known to exist, and (3) a horizontal scan of port 3389 on 3,389,187 IP addresses, 12,405 of which were known to exist. The one false positive consisted of 47 SYN-only flows to five IP addresses, with each IP address contacted on a different, high-numbered port (specifically, ports 1166, 19592, 27947, 43525, and 61519). This traffic is certainly unusual, but the expert analyst felt that it represented some other type of activity other than a scan. The TRW algorithm, however, performed much better on the testing set, correctly identifying all of the scanners with no false positives. Interestingly, it did not identify any of the non-scanners as benign, but rather all of them were still in a state pending a decision as to whether they represented scanning or non-scanning activity.

It should be noted at this point that the comparisons made above are based on source IP addresses that are associated with at least 32 flows. One of the advantages to using TRW is that it can accept or reject the hypothesis that a source is scanning given very few connection attempts (the average case given the parameters from the paper is 5.4 connection requests performed before the hypothesis is accepted or rejected [6]). Thus for any scans consisting of fewer than 32 flows, TRW will outperform our detection approach. The value of 32 flows was chosen arbitrarily for our data, and in the future we intend to investigate the minimum number of flows required to detect a scan using our model.

6 Conclusions and Future Work

In this paper we presented a new approach to scan detection. Unlike previous approaches, ours requires only unidirectional flow data and is thus suitable for very large networks such as ISPs, large corporations and government organizations. Our approach is based on the concept of analysing all of the traffic from each source over a particular time period. Several key characteristics of the traffic are extracted and a logistic regression model is applied to determine the probability that the traffic contains scanning activity. This approach is currently in operational use on an ISP network, processing more than 10 Tb of data across more than two billion flows per day.

Three hundred events were randomly chosen from a pool of over 100,000 TCP events, and these were manually characterized by an expert analyst into scans and non-scans. The results from the logistic regression model were then compared against this assessment. We compared our approach to a modified threshold random walk (TRW) [6] using the same training and testing sets. We had a classification accuracy of 98.5% on the training set and 99.3% on the testing set, while TRW's accuracy was 98.0% and 100.0% respectively. Thus our results are comparable.

References

[1] H. Akaike. Information theory as an extension of the maximum likelihood principle. In B. N. Petrov and F. Csaksi, editors, *Proceedings of the 2nd International Symposium on Information Theory*, pages 267 – 281, Budapest, Hungary, 1973.

[2] Stefan Axelsson. The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security*, 3(3):186 – 205, 2000.

[3] G. Elfving. Optimum allocation in linear regression theory. *The Annals of Mathematical Statistics*, 23(2):255 – 262, 1952.

[4] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall, 1995. ISBN 0-412-03991-5.

[5] Simon Jackman. Estimation and inference via bayesian simulation: An introduction to markov chain monte carlo. *American Journal of Political Science*, 44(2):375 – 404, 2000.

[6] Jaeyeon Jung, Vern Paxson, Arthur W. Berger, and Hari Balakrishnan. Fast portscan detection using sequential hypothesis testing. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, pages 211 – 225, Oakland, California, USA, 2004. IEEE Computer Society. May 9-12, 2004.

[7] C. Leckie and R. Kotagiri. A probabilistic approach to detecting network scans. In *Proceedings of the 2002 IEEE Network Operations and Management Symposium*, pages 359 – 372, Florence, Italy, 2002. April 15-19, 2002.

[8] Chris Muelder, Kwan-Liu Ma, and Tony Bartoletti. Interactive visualization for network and port scan detection. In *Proceedings of 2005 Recent Advances in Intrusion Detection*, 2005. September 7-9, 2005.

[9] Susmit Panjwani, Stephanie Tan, Keith M. Jarrin, and Michel Cukier. An experimental evaluation to determine if port scans are precursors to an attack. In *Proceedings of the 2005 International Conference on Dependable Systems and Networks*, pages 602 – 611, Yokohama, Japan, 2005. June 28-July 1, 2005.

[10] Vern Paxson. Bro: A system for detecting network intruders in real-time. In *Proceedings of the 7th USENIX Security Symposium*, 1998. San Antonio, Texas. January 26-29.

[11] Seth Robertson, Eric V. Siegel, Matt Miller, and Salvatore J. Stolfo. Surveillance detection in high bandwidth environments. In *Proceedings of the 2003 DARPA DISCEX III Conference*, pages 130 – 139, Washington, DC, 2003. IEEE Press. 22-24 April 2003.

[12] Martin Roesch. Snort — lightweight intrusion detection for networks. In *Proceedings of LISA '99: 13th Systems Administration Conference*, 1999. Seattle, Washington, USA, November 7-12, 1999.

[13] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle. GrIDS – a graph based intrusion detection system for large networks. In *Proceedings of the 19th National Information Systems Security Conference*, 1996. Baltimore.

[14] Nicholas Weaver, Ihab Hamadeh, George Kesidis, and Vern Paxson. Preliminary results using scale-down to explore worm dynamics. In *Proceedings of the 2004 ACM Workshop on Rapid Malcode*, pages 65 – 72, Washington, DC, 2004. October 29, 2004.

[15] Nicholas Weaver, Stuart Staniford, and Vern Paxson. Very fast containment of scanning worms. In *Proceedings of the 13th USENIX Security Symposium*, pages 29 – 44, San Diego, CA, 2004. August 9-13, 2004.

[16] Sanford Weisberg. *Applied Linear Regression*. Wiley Series in Probability and Mathematical Statistics, second edition, 1985. ISBN 0271-6356.